P1

a. $f(n) = 12n-5$, $g(n) = 1235813n + 2017$.

For O(g(n)), find c, n0,
make f(n) <= cg(n),
12n-5 <= 1235813n+2017,
(12-1235813c)n <= 2017c +5,
n<=(2017c+5)/(12-1235813c)
hence, contradiction as n cannot hold all n>n0, f is not order of g(n).

for omega(g(n)), find c, n0,
make f(n) >= cg(n),
same with above,
n >=(2017c+5)/(12-1235813c),
hence, exist n0 makes f(n) is omega of g(n).

b. $f(n) = n\log n$, $g(n) = 0.00000001n$.
assume the base is a,
For O(g(n)), find c, n0,
nlogn<= 0.00000001nc,
logn<= 0.00000001c,
n<=a^(0.00000001c),
hence, contradiction as n cannot hold all n>n0, f is not order of g(n).

for omega(g(n)), find c, n0,
same with above,
n>=a^(0.00000001c),
hence, exist n0 makes f(n) is omega of g(n).

c. $f(n) = n^{2/3}$, $g(n) = 7n^{3/4} + n^{1/10}$.
For O(g(n)), find c, n0,
n^(1/2)-cn^(2/15)<=7c,
for n^(1/2)-cn^(2/15) part, assume c = 1,
n^(1/2)-n^(2/15)<=7,
The derivation of the left side is 1/2 * n ^(-1/2) – 2/5 *n^(-13/15), Is 15/(30*n^(15/30)) - 4/(30*n^(26/30)),
It's bigger than 0 when n is sufficiently large, which means value of the left side will increase,
Let's set the n = 10^10, the result is larger than 7,
hence, when n is big, there has a condition that does not suit for the function, contradiction as n cannot hold all n>n0, f is not order of g(n).

for omega(g(n)), find c, n0,
same with above,
n^(1/2)-cn^(2/15)>=7c,
the left side is increasing and there exist n's that let it larger than 7,
hence, exist n0 makes f(n) is omega of g(n).

d. f(n) = n^1.0001, g(n) = nlogn.
For O(g(n)), find c, n0,
1/c <= n^(-0.0001)logn, noted as (1), assume the base is a,
Assume c = 10,000, a =10, when n = 100,000,000, the function works,
1/10000 <= 1/10,000 * 8
For the right side of (1), the derivation for it is n^(-1/10000) * 1/(n*ln10) - 1/10000 * n^(-10001/10000)*logn, both of them are larger than 0 when n is large enough, I don't know where the result of the derivation might below the 0, which means the value of (1) may decrease till smaller than 1/c, so complicated by hands.

Hence, I don't know.

e. f(n) = n6^n, g(n) = (3n)^2.
For O(g(n)), find c, n0,
n*(2/3)^n<=c,
The derivation of the left side is n * (2/3)^n * ln(2/3) + (2/3)^n, larger than 0 when n is big enough. Hence, n*(2/3)^n will larger than c somehow.
Hence, contradiction as n cannot hold all n>n0, f is not order of g(n).

Reversely, for omega g(n),
exist n0 makes f(n) is omega of g(n).

P2
Log(n!) = theta (nlogn), base: 2
f(n)=log(n!), g(n)=nlogn,

For O(g(n)), find c, n0,
Log(n!) = logn + log(n-1) + …… + log(1) <= log n + …… + logn = nlogn,
Hence, f(n) <= cg(n), f is order of g(n).

for omega(g(n)), find c, n0,
log(n!) = logn + log(n-1) + …… + log(1) >= n/8 * log(n/8) >= c logn,
Then we need to find at least a “c”,

n>=16,
logn>=4,
1/16 * nlogn >= 1/4 * n,
1/8 * nlogn – 1/4 * n >= 1/16 * nlogn, all positive,
Hence, when c = 1/8, and n> n0 = 16, log(n!) is larger than c logn,
Which means f(n) is omega of g(n).

Hence, f(n) is theta of g(n).

P3

```python

def trans(x):
    if x == 1 or x == 0:
        return x
    else:
        print(x%2)
        return trans (x//2)

print(trans(5))
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL 1: Python

```
+ conda activate base
+ ~~~~~
    + CategoryInfo          : ObjectNotFound: (conda:String) [], CommandNotFoundException
    + FullyQualifiedErrorId : CommandNotFoundException

PS C:\Users\GR> & C:/Users/GR/Anaconda3/python.exe d:/GRADUATE_OSU/Fall_2020/CS325_retaken/pa1_3.py
PS C:\Users\GR> & C:/Users/GR/Anaconda3/python.exe d:/GRADUATE_OSU/Fall_2020/CS325_retaken/pa1_3.py
1
0
1
```

P4

Starting from the root node, and visit the left node of itself. Creating a new small tree and treat the node we mentioned as a new root. Keep doing that until the bottom left leaf. Then visit right leaf and back.

```
              1
     2                  5
3         4         6         7
```

Postorder would find the most left and bottom one first and the right node match with it, then back to the subroot node.

```
              7
     3                  6
1         2         4         5
```